

新世纪高职高专规划教材

汇编程序设计

主 编	辽宁机电职业技术学院	赵景晖
副主编	北京城市学院	杨锡平
参 编	西安理工大学高等技术学院	杨景林
	辽宁机电职业技术学院	孙莉娜
	安徽商贸职业技术学院	秦体德

机械工业出版社

新世纪高职高专规划教材

编委会名单

编委会主任 王文斌 郝广发

编委会副主任 (按姓氏笔画为序)

马元兴	王茂元	王明耀	王胜利	王锡铭
田建敏	刘锡奇	杨文兰	杨 飒	李兴旺
李居参	杜建根	余元冠	沈国良	沈祖尧
陈丽能	陈瑞藻	张建华	茆有柏	徐铮颖
符宁平	焦 斌			

编委会委员 (按姓氏笔画为序)

王志伟	付丽华	成运花	曲昭仲	朱 强
齐从谦	许 展	李茂松	李学锋	李连邨
李超群	杨克玉	杨国祥	杨翠明	吴诗德
吴振彪	吴 锐	肖 珑	何志祥	何宝文
陈月波	陈江伟	张 波	武友德	周国良
宗序炎	俞庆生	恽达明	娄 洁	晏初宏
倪依纯	徐炳亭	唐志宏	崔 平	崔景茂

总 策 划 余茂祚

策划助理 于奇慧

前 言

本书是根据教育部教高〔2006〕16号文件精神，由中国机械工业教育协会和机械工业出版社组织全国 100 多所高等院校合作编写的高职高专规划教材之一。

互联网技术在全球范围内突飞猛进地发展，给计算机的应用领域带来了巨大的变化，计算机应用的网络化已经成为发展的必然趋势。作为网络编程语言，Java 已经受到越来越多 的行业科研人员的重视，其应用领域也越加广泛。因此，很多高等院校在相关专业相继开设了 Java 程序设计的课程。

Java 是纯面向对象的程序设计语言，其良好的跨平台特性和安全机制使得它特别适合开发网络应用程序。Java 是一种程序设计语言，更是一种构建互联网应用开发的工具和解决方案。作为信息类专业的学生，掌握 Java 编程技术具有十分重要的意义。

本书从 Java 的基本特点开始，逐步介绍了 Java 的基础知识，面向对象编程的概念和特点，面向对象编程的高级特性等。在此基础上，对 Java 的数组和字符串、异常处理机制、图形界面、输入输出与文件管理、线程、数据库编程和网络通信编程等进行了详细的阐述。为提高读者运用 Java 解决实际问题的能力，在每章中安排了较多的应用实例。本书在内容阐述方面，尽量做到语言浅显易懂；书中给出的实例保持风格一致，代码书写规范。

根据教育部有关文件精神及高职高专发展趋势，本教材基本上是按照高中毕业二年制高职高专教学计划所需内容编写的，同时兼顾三年制高职高专的教学计划。教学内容以工程应用为目的，以理论适度、讲清概念、强化应用为重点，突出实用性、综合性，注重学生基本技能的训练和综合能力的培养。本书可作为高等职业技术学院、高等专科学校、职工大学、业余大学、成人教育学院、电视大学以及其他大专层次的信息类专业的教材，也可供有关教师以及工程技术人员参考。

本教材共 18 章，其中第 1 章～第 10 章由辽宁机电职业技术学院赵景晖编写；第 11 章～第 14 章由辽宁机电职业技术学院孙莉娜编写；第 15 章～第 17 章由安徽商贸职业技术学院秦体德编写；第 18 章由北京城市学院杨锡平编写；第 19 章由西安理工大学高等技术学院杨景林编写。本书由赵景晖任主编，杨锡平任副主编。

本书在编写过程中得到辽宁机电职业技术学院领导的大力支持和热情帮助，在此也表示感谢。

由于编者水平有限，加之时间仓促，书中难免有错误和疏漏之处，恳请读者给予批评指正。

目 录

前言		绪论 捕获异常	vi
第 1 章 介绍性语言概述	1	绪论 丢弃异常	vi
1.1 介绍性语言的发展及特点	1	绪论 抛出异常	vi
1.2 介绍性虚拟机	1	绪论 异常类常用的方法	vi
1.3 简单的介绍性程序	1	绪论 自定义异常	vi
1.4 介绍性程序的编译与运行	1	复习思考题	vi
1.5 常用开发工具介绍	1	第 2 章 数组与字符串	1
复习思考题	1	2.1 数组	1
第 2 章 介绍性基本语法	1	2.2 字符串	1
2.1 介绍性字符	1	复习思考题	1
2.2 数据类型、常量和变量	1	第 3 章 图形界面设计	1
2.3 运算符、表达式和语句	1	3.1 图形用户界面的基础知识	1
2.4 语句和程序流程控制	1	3.2 常用组件	1
复习思考题	1	3.3 布局管理	1
第 3 章 类和对象的基础知识	1	3.4 事件处理	1
3.1 面向对象程序设计的概念	1	3.5 图形图像处理	1
3.2 类的概念	1	3.6 杂项组件简介	1
3.3 类的定义	1	复习思考题	1
3.4 对象的创建及类成员的引用	1	第 4 章 介绍性粤景编程	1
3.5 类的成员	1	4.1 粤景编程基础	1
3.6 类的继承	1	4.2 粤景编程类	1
3.7 包 (粤景编程)	1	4.3 简单粤景编程程序举例	1
复习思考题	1	4.4 粤景编程标记语法	1
第 4 章 类的高级特性和接口	1	4.5 粤景编程和粤景编程的参数传递	1
4.1 方法的覆盖	1	4.6 在粤景编程中显示图像	1
4.2 多态性	1	4.7 在粤景编程中播放声音	1
4.3 抽象类和最终类	1	4.8 粤景编程响应鼠标事件	1
4.4 接口	1	4.9 粤景编程响应键盘事件	1
4.5 内部类	1	复习思考题	1
4.6 匿名类	1	第 5 章 输入输出与文件管理	1
复习思考题	1	5.1 流的概念	1
第 5 章 异常处理	1	5.2 流的分类及层次	1
5.1 异常的概念	1	5.3 字符流类	1
5.2 异常的分类及层次	1	5.4 字节流类	1

第 10 章 文件管理与操作	100	第 10 章 网络基础	100
第 10.1 节 文件的随机读写	100	第 10.1 节 网络包 (网络协议)	100
第 10.2 节 文件流	100	第 10.2 节 使用 socket 协议编程	100
第 10.3 节 对象的序列化	100	第 10.3 节 使用 socket 协议编程	100
复习思考题	100	第 10.4 节 使用 socket 类编程	100
第 11 章 多线程程序设计	100	复习思考题	100
第 11.1 节 线程的概念	100	第 11 章 数据库编程	100
第 11.2 节 简单的多线程机制	100	第 11.1 节 数据库的概念	100
第 11.3 节 线程的运行状态	100	第 11.2 节 数据库驱动程序的类型	100
第 11.4 节 线程的创建方法	100	第 11.3 节 数据库简介	100
第 11.5 节 线程的优先级	100	第 11.4 节 创建与数据库的连接	100
第 11.6 节 多线程和线程同步	100	第 11.5 节 访问数据库	100
第 11.7 节 线程的调度	100	复习思考题	100
复习思考题	100	附录 安装和配置 MySQL 数据库	100
第 12 章 网络编程	100	参考文献	100

第 1 章 JavaScript 语言概述

本章主要介绍 JavaScript 语言的发展概况、技术特点和 JavaScript 虚拟机的概念，用几个简单的例子说明 JavaScript 程序的类型、结构以及编译和运行 JavaScript 程序的方法。为使初学者能够以较高的效率开发 JavaScript 应用程序，还介绍了 JavaScript 辅助开发工具 NetBeans 的使用方法。

1.1 JavaScript 语言的发展及特点

1.1.1 JavaScript 语言的发展

JavaScript 语言诞生于 20 世纪 90 年代初，它是由美国 Netscape 公司的一个开发小组在开发 Navigator 项目时完成的，该项目旨在为消费类电子产品开发一种交互式的软件系统。最初该小组尝试用 Java 作为编程语言，但在项目进行过程中，发现 Java 语言存在编程复杂、可移植性差等缺点。于是选择基于 Java 开发一种新的计算机编程语言，该项目小组将之命名为 LiveScript。LiveScript 继承了 Java 面向对象等优点，摒弃了 Java 编程复杂、可移植性差等缺点。在 Netscape 公司正式为该语言注册时，发现 LiveScript 已经被其他产品注册了，于是将该语言正式命名为 JavaScript。

JavaScript 诞生之初，并未引起计算机界专业人士的重视，直到 20 世纪 90 年代中期，随着 Internet 的蓬勃发展，受 Java 和 JavaScript 的启发，该项目小组用 JavaScript 开发了浏览器软件 Navigator。该浏览器可以运行用 JavaScript 编写的小程序，实现了在 Web 页中进行交互操作的功能。Navigator 优异的表现为 JavaScript 赢得了良好的声誉，使计算机界专业人士认识到 JavaScript 的巨大应用潜力和发展前景。如今，许多世界著名的计算机软硬件公司如 IBM、微软等相继购买了 JavaScript 语言的使用许可，越来越多的软件和硬件产品将与 JavaScript 技术结合，JavaScript 已经成为网络、多媒体等编程的强大工具。

1.1.2 JavaScript 语言的特点

同其他程序设计语言相比，JavaScript 技术的发展十分迅速，其应用也越来越广泛，这与 JavaScript 所具有的突出特点密切相关。归纳起来，JavaScript 的主要特点如下：

1. 简单性 JavaScript 是面向对象的编程语言，在语法规则上和 Java 类似，Java 程序员会比较容易地掌握 JavaScript 编程技术。JavaScript 摒弃了 Java 容易引起错误的功能，如指针操作和内存管理。JavaScript 本身提供了丰富的类库，程序员可以直接使用类的对象实现较为复杂的程序设计任务，使程序设计变得简单、高效。

2. 面向对象 面向对象是 JavaScript 的重要特性。面向对象编程技术具有很多优点，比如通过对象的封装，减少了对数据非法操作的风险，使数据更加安全；通过类的继承，实现了代码的重用，提高编程效率等。JavaScript 语言的编程主要集中在类、接口的描述和对象引用方面，面向对象编程技术适合开发大型、复杂的应用程序，且程序代码易于理解和维护。

3. 分布式 JavaScript 是面向网络的编程语言，它提供了基于 HTTP 协议的类库，使用这些类，JavaScript 应用程序可以很容易地访问网络上的资源。编译后生成的目标代码可以分布在网络的任何主机上，客户机能够方便地实现和这些目标程序的通信。

4. 健壮性 JavaScript 提供了较完备的错误处理机制，在编译和运行程序时，系统对代码进行逐级检查，指出可能产生错误的地方，要求必须对可能存在错误的代码进行必要的处理，以

圆

消除因产生错误而造成系统崩溃的情况。

安全性 作为网络编程语言，安全是至关重要的。一方面，由于 **Java** 不支持指针，消除了因指针操作带来的安全隐患；另一方面，**Java** 具有完备的安全结构和策略，代码在编译和运行过程中，被逐级检查，可以防止恶意程序和病毒的攻击。另外，对下载到本地的 **Java** 类，严格地限制其访问本地资源的权限，降低了利用 **Java** 非法访问本地资源的风险。

平台独立与可移植性 互联网是由各种各样的计算机系统组成的，如果应用程序在网络中任何计算机上都能够正常运行，必须使程序具有平台独立性，即软件本身不受计算机硬件和操作系统的限制。**Java** 是一种具有平台独立性的编程语言，**Java** 源程序编译后生成与具体机器指令无关的字节码，通过 **Java** 虚拟机（**JVM**）在不同的平台上运行。**Java** 的基本数据类型在设计上不依赖于具体硬件，为程序的移植提供了方便。

解释执行 **Java** 是解释执行的编程语言，**Java** 源程序经过编译后生成被称作字节码（**Bytecode**）的二进制文件，**Java** 解释器直接执行字节码文件。解释器在执行字节码文件时，能够对代码进行安全检查，以保证没有被修改的代码才能执行，提高了系统的安全性。

多线程 多线程机制使程序代码能够并行或重叠执行，而且线程的同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以用不同的线程完成不同的行为，而不需要采用全局的事件循环机制，这样就实现了网络上的实时交互行为。

动态性 **Java** 在设计上力求适合不断发展的环境。在类库中可以自由地加入新的方法和实例而不会影响用户程序的执行。**Java** 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式且易于扩展。

2.1 **Java** 虚拟机

Java 虚拟机（**JVM**）是在实际计算机上用软件实现的计算机模型，它有自己的虚拟的处理器、堆栈、寄存器等，还具有相应的指令系统。**Java** 语言的一个非常重要的特点就是平台独立性，使用 **Java** 虚拟机是实现这一特点的关键所在。

对一般的高级语言而言，如果要在不同的平台上运行，要用适用于不同平台的编译器对源程序进行重新编译，生成不同的目标代码，比如 **C** 语言编译系统下的 **gcc** 编译器生成的是 **x86** 处理器上的二进制代码。**Java** 编译器生成的目标代码不是面向具体处理器的代码，而是面向虚拟机的中间代码，称为字节码（**Bytecode**），字节码只能在 **Java** 虚拟机上运行。

目前很多计算机系统都支持 **Java** 虚拟机。**Java** 虚拟机概括起来有三部分组成：类下载器、字节码校验器和执行单元。图 2-1 是 **Java** 编译器、虚拟机和计算机系统之间的关系。

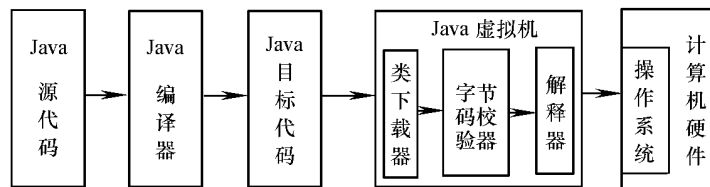


图 2-1 **Java** 编译器、虚拟机和计算机系统之间的关系

虚拟机用类下载器从计算机或网络上获取字节码，字节码经过字节码校验器验证之后，

远

用于 **Java** 开发的辅助工具，如 **Ant**、**Eclipse**、**NetBeans**、**MyEclipse** 等。

集成开发工具在很大程度上扩展了 **Java** 的功能，它可以生成各种应用程序的框架，提供可视化程序设计环境，使程序设计变得简单、高效。但多数集成开发工具需要它自己提供的类库的支持，这是 **Java** 初学者难以掌握的内容。

一般的辅助开发工具，能够把程序的编辑、编译和运行集成在一起，尤其在程序的编辑过程中，可以自动显示 **Java** 提供的包（**API**）中的类名称、类的方法的名称等，提高了编程效率。有些辅助工具是免费的，如 **NetBeans**、**Eclipse** 等。初学者可以在熟悉 **Java** 工具后，选择一个辅助开发工具，以提高学习效率。

1.1.1 **Java** 开发工具

Java 是 **Sun** 公司免费提供的开发工具，常见的版本有 **Java 2**、**Java 5**、**Java 6** 等。可以从 **Sun** 的网站（<http://www.java.com>）上下载适合各种平台的 **Java** 工具包。在 **Windows** 平台下，下载名为 **jdk-6u0-windows-i586.exe** 的文件。该文件为自解压文件，将文件安装在适当的目录下，如“**C:\Program Files\Java**”，安装后的目录结构见图 1-1。



图 1-1 安装 **Java** 后生成的目录结构

各目录存放的文件及用途如下：

- bin** 存放编译、运行、调试等系统工具；
- include** 存放与 **Java** 相关的头文件；
- lib** 存放与 **Java** 运行环境相关的文件；
- src** 存放 **Java** 标准类库；
- demo** 存放系统提供的应用实例。

在编译、运行和调试 **Java** 程序时，所使用的工具存放在 **bin** 目录下，这些工具的名称和作用如下：

- javac** **Java** 编译程序，将 **Java** 源程序编译成字节码文件。
- java** **Java** 解释程序，运行字节码文件。
- appletviewer** **Java** 浏览器，执行包含 **Java** 的 **HTML** 文件。

根据源代码即说明语句生成文档。

使用调试器，可逐行执行程序，设置断点和检查成员变量。

使用反汇编工具，显示字节码文件中的可访问方法和数据。

在安装好系统以后，为了编译、运行和调试程序的方便，应当设置系统的环境变量。一方面，可以在任何目录下执行编译、运行和调试程序时，不用输入这些程序的路径，另一方面，在执行类文件或字节码文件时，也可以不用指出文件的路径。如果使用系统，可以利用文本编辑工具，在系统文件中加入“系统”的内容。如果执行的类文件不在当前目录下，还应当设置系统参数，比如类文件分别在“系统”和“系统”目录中，可以在系统文件中加入“系统”和“系统”，以便虚拟机能够找到相应的类文件。

对于系统，需要按以下方式设置系统的环境变量：

启动控制面板→选择“系统”→选择“高级”，出现如图 10-1 的界面。

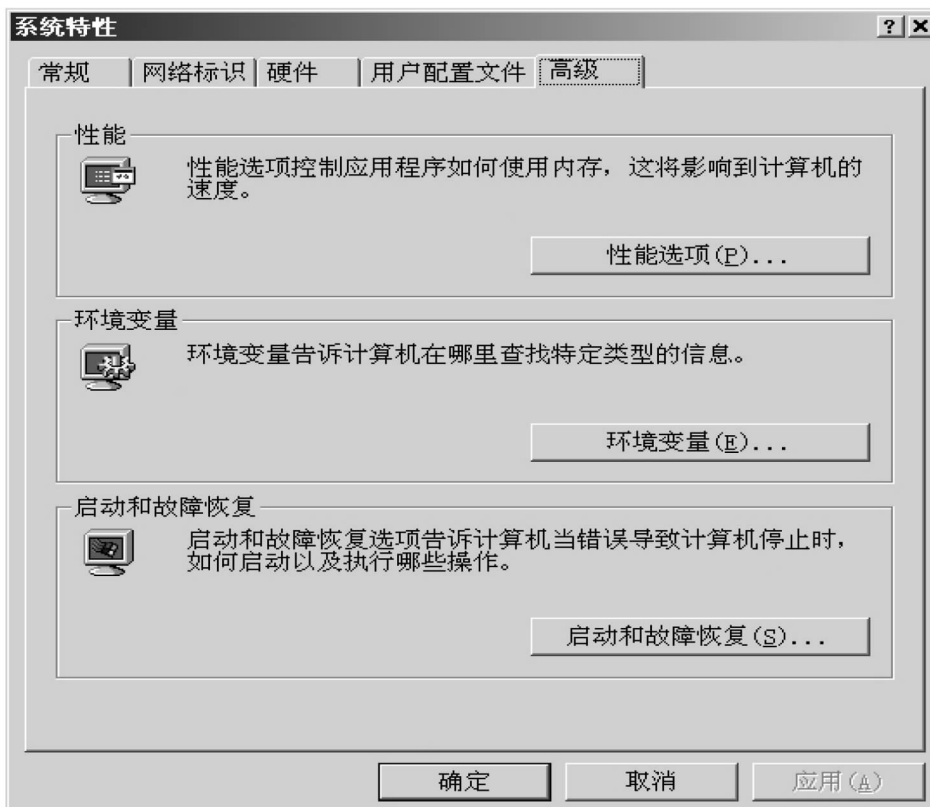


图 10-1 环境变量设置之一

在图 10-1 中选择环境变量，出现图 10-2 的界面。

在图 10-2 中的“系统变量”窗口中，利用窗口右侧的滚动条找到“系统”项目，选中“系统”，出现如图 10-3 的界面。

愿

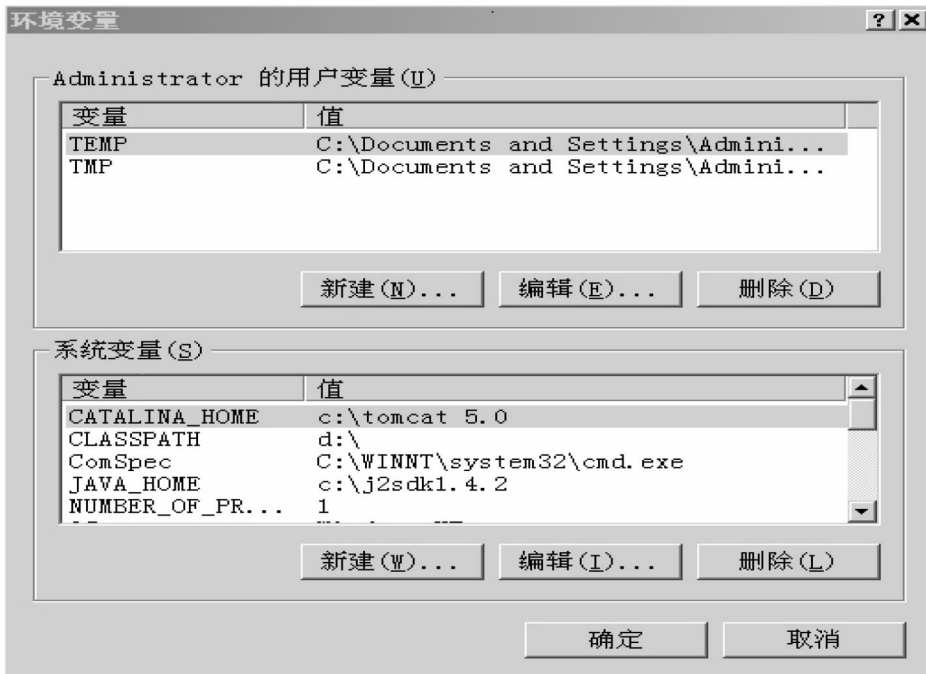


图 10-10 环境变量设置之二



图 10-11 环境变量设置之三

图

图 辅助开发工具

是由 公司免费提供的共享软件，该软件可以从 网站上下载，在 系统下，下载名为 的文件，该文件是一个自解压的文件，可以直接安装到用户的系统中。的开发者们正在不断地更新该软件，目前会有更新的版本。使用 可以完成 程序的编辑、编译和运行。下面，介绍如何使用 来编辑、编译和执行 程序。图 是 运行后的工作界面。

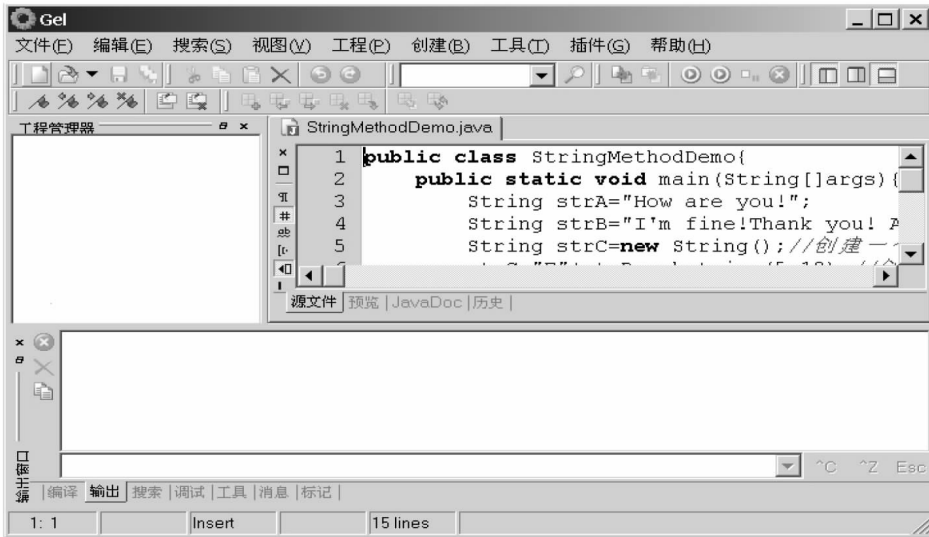


图 应用程序界面

运行环境的设置 是一个辅助开发工具，该软件需要调用 来完成程序的编译和运行，因此在使用 之前，应当设置 的运行环境，以确定编程风格、 中应用程序的位置（路径）、类库的位置等。

在 的主菜单上，“工具（裁）”菜单中的“选项（韵）”项目，用于设置 的环境参数。用鼠标点击该项目后， 的界面如图 所示。界面分两部分：左面是一个树型项目列表，右面是以页面形式给出的参数设置列表。 除了能够用于 程序的编辑和调试外，也可以用于其他语言如 等的编辑与调试。在选项列表中，选择与 相关的项目，设置 程序的编辑环境，例如在输入左“{”时，让编辑器自动加上右“}”，让系统保留字、变量等以不同的颜色显示出来等等。因为选项较多，可以根据程序的编辑风格，选中需要的项目，合理的设置编辑风格会给程序的阅读和编辑带来很大的方便。

在这里必须配置的是 选项，因为 需要执行 提供的编译、调试、运行等程序，编译时还需要调用 提供的类库。该项目如果配置不正确， 便不能正确完成编译、调试、运行程序等功能。图 是配置 编译程序的界面，通过配置该项目， 可以到指定的路径中找到相应的编译程序。具体方法是：在图 右侧的页框中，选择编译标签，再在下面编译对应的文件栏目中，输入“ 所在的路径，如图 中输入的“糟 则，在执行编译时， 自动到相应的路径中执行编译程序。用

同样的方法，可以设置 解释程序（[跟踪器](#)、[解释器](#)、[编译的调试程序](#)（[跟踪器](#)）和小应用程序观察器（[类库](#)、[类库](#)、[类库](#)）等的查找路径。

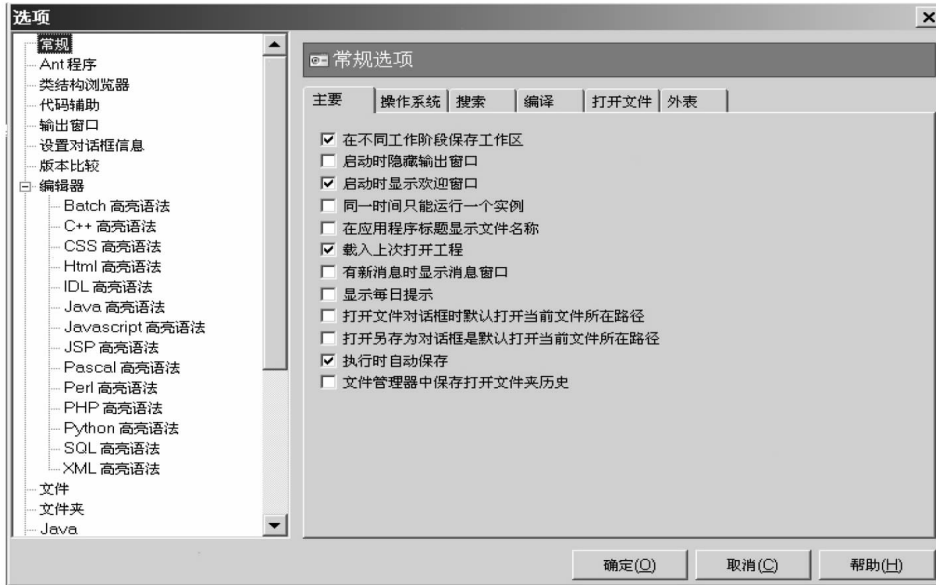


图 环境设置界面



图 编译程序的界面

提供的类库，均发布在 程序包中，该文件存放在“[输入](#)、[输出](#)、[类库](#)”路径下，在类路径中必须指出这个文件所在的位置，设置该项目的界面如图 所示。设置方法是：选择图 中的“类路径”标签，出现图 的界面，在该界面中，选择新增文件选项，在随后出现的文件选择对话框中，选择“[输入](#)、[输出](#)、[类库](#)”路径，并在该路径